

## REMARKS

This amendment is submitted in response to a final Office Action mailed January 13, 2005. Claims in their pending forms are also submitted herein. As the Examiner has dismissed Applicant's arguments from the previous Response, additional arguments are presented below to specifically address the Examiner's response to Applicant's arguments.

1. Applicant respectfully submits that the Examiner has failed to prove that a device driver is the same as a link driver.

Examiner has not shown in Shima any language that supports a finding that a device driver and a link driver are equivalent. To the contrary, the Examiner cited a different patent, White (U.S. patent 6,041,286) to support such a finding.

A. MPEP 2136.02 Content of the Prior Art Available Against the Claims states that a reference must itself contain the subject matter relied on in the rejection. Shima, which is the prior art cited against Applicant's claims, does not contain the subject matter relied on in the rejection. Furthermore, White is not the art relied on in the rejection.

B. Even if White WAS the reference relied on in the rejection, it does not contain subject matter that supports the 102(e) rejection. Examiner has cited White at col. 4, lines 8-20 as disclosing that device drivers control a link chip; in other words, the device drivers include a link driver for controlling the link chip.

Applicant respectfully asserts that White contains no such disclosure. White, at Col. 4, lines 8-20, reads:

*A link chip 52 is coupled to the physical transceiver circuit 42 for providing data and control signals from device drivers and applications to the physical transceiver circuit 42. The link chip 52 is included within the interface circuit 28. **The software applications and device drivers communicate with the link chip 52.** The relevant software applications and device drivers for transmitting data from the node over the IEEE 1394 serial bus network include the IEEE 1394 port driver 54, the IEEE 1394 bus class driver 56 and the digital video mini driver 58. The drivers 54, 56 and 58 reside within the operating system and provide the instructions and data necessary to transmit a video frame.*

Examiner's attention is drawn to the cited portion of White, "**The software applications and device drivers communicate with the link chip 52.**" Saying that a device driver communicates with a link chip is not the same as saying that a device driver controls a link chip. Similarly, while a client *communicates* with a server, it cannot be said that a client *controls* the server.

2. Shima does not teach disclose or otherwise suggest generating a link driver configuration for the link driver from the capabilities received as claimed in claims 10 and 21.

Examiner states that an appropriate device driver corresponding with a device of a plurality of devices should be identified and loaded in the processor in order to control the link chip of IEEE 1394-1995 link specifics and cites Shima at Col. 2, lines 31-39 as disclosing this point. However, Shima at Col. 2, lines 33-39 reads:

*The serial bus management block 10 also provides overall configuration control of the serial bus in the form of optimizing arbitration timing, guarantee of adequate electrical power for all devices on the bus, assignment of the cycle master, assignment of isochronous channel and bandwidth resources and basic notification of errors.*

However, this portion of Shima does not disclose, teach or suggest generating a link driver configuration; rather it specifically focuses on optimizing arbitration timing, guarantee of adequate electrical power for all devices on the bus, and assigning channel and bandwidth resources. There is no mention of link drivers or how link drivers may be configured. As such, this cited portion cannot be construed to disclose generating a link driver configuration.

3. Shima does not teach disclose or otherwise suggest loading the link driver configuration in the link driver as claimed in claims 10 and 21.

Applicant respectfully submits that Examiner has failed to address this argument, and mischaracterized the argument as stating only that Shima is talking about generating objects that represent devices, not detecting capabilities of a link driver. Examiner's attention is drawn again to the limitation at issue, of *loading the link driver*

*configuration in the link driver.* Applicant's support for this argument, yet unaddressed by the Examiner, is again submitted below:

Examiner has responded to this argument made by Applicant by stating that Shima teaches that the link driver configuration should be loaded into the link driver to reflect or reconfigure the bus changes or updates [col. 3, lines 22-37; col. 5, lines 41-47] so that the link driver is able to access the device with the capabilities determined [col. 4, lines 1-10]. However, this argument is not supported by the cited portions of Shima.

Shima, at col. 3, lines 22-37 reads:

*A reset event signifies to the controlling or monitoring application that the status of the bus and the nodes connected to it has changed. This requires the controlling or monitoring application to somehow update its information regarding the devices connected to the serial bus network. This can be a significant endeavor. There is currently a lack of efficient apparatus and methods for updating information for nodes on a serial bus network after a bus reset. No solutions for handling the updating of device information after a bus reset is described in any of the IEEE 1394 standards documents.*

#### SUMMARY OF THE INVENTION

A controlling application utilizes existing handle objects, as appropriate, to reconfigure objects to dynamically enumerate and represent devices coupled to a serial bus network after a bus reset event. Preferably, the serial bus network is an IEEE 1394-1995 serial bus network.

There is no mention of a link driver anywhere in this text. Furthermore, Shima at Col. 3, lines 12-16 clarifies what is meant by the object that is maintained by a controlling or monitoring application:

*"Generally, such a controlling or monitoring application maintains a representation or object of each device. This object represents the capabilities of the device."*

Again, Applicant asserts that a DEVICE is not a LINK DRIVER. This cited portion of Shima does not teach loading a LINK DRIVER generated as claimed in claims 10 and 21 of the present application.

Shima at col. 5, lines 41-47 reads:

*"A bus reset occurs when the configuration of the nodes within the IEEE 1394-1995 serial bus network changes. Accordingly, the objects and handles representing devices within the network must somehow be updated after a bus reset to reflect the current state of the nodes coupled to the IEEE 1394-195 serial bus network."*

Again, there is no discussion of a link driver in this cited portion of Shima. An object represents a device. A device is not a link driver. This cited portion does not disclose, teach, nor otherwise suggest loading the link driver configuration in the link driver as claimed in claims 10 and 21.

Shima at col. 4, lines 1-10 reads:

*"...of maintaining a library of a plurality of available subobjects each representing an available subunit, determining characteristics of the device, including resident subunits within the device, retrieving retrieved subobjects from the library corresponding to the resident subunits and assembling the retrieved subobjects into an object representing the functions and characteristics of the device. The method further comprises the step of receiving self identifying information for the device. The characteristics of the device are determined from the self identifying information.*

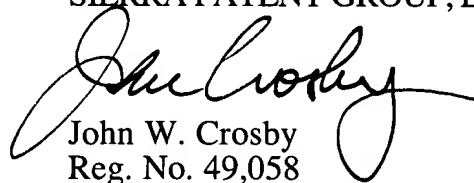
Again, Applicant asserts that the Examiner has once again mischaracterized the prior art, and this portion in no way teaches, suggests, nor otherwise discloses loading the link driver configuration in the link driver as claimed in claims 10 and 21. Objects represent devices. Devices are not drivers. Objects do not represent link drivers. Shima does not even once discuss a link driver configuration. NO part of Shima relates to loading a link driver configuration into a link driver.

**If the Examiner wishes to maintain a rejection under 35 USC 102 or 35 USC 103 of this claim limitation, Applicant demands proof that the cited prior art teaches, suggests, or otherwise discloses it.**

Dated: March 14, 2005

Sierra Patent Group, Ltd.  
PO Box 6149  
Stateline, NV 89449  
(775) 586-9500

Respectfully submitted,  
SIERRA PATENT GROUP, LTD.



John W. Crosby  
Reg. No. 49,058